UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/764,526 | 01/17/2001 | Christopher R. Sheedy | 20206-39 (P00-3337) | 6417 |

7590 11/26/2003

HEWLETT-PACKARD COMPANY
Bill Streeter
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

| EXAMINER | |
|---|---|
| CHOW, CHIH CHING | |
| **ART UNIT** | **PAPER NUMBER** |
| 2122 | 6 |

DATE MAILED: 11/26/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/764,526 | SHEEDY, CHRISTOPHER R. | |
| | Examiner | Art Unit | |
| | Chih-Ching Chow | 2122 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *17 January 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☐ Claim(s) _____ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *04/30/2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. §§ 119 and 120**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

## DETAILED ACTION

### *Claim Rejections - 35 USC § 102*

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

2. Claims 1 and 7 are rejected under 35 U.S.C. 102(e) as being anticipated by Boehm et al, US Patent No. US6457170 (herein after Boehm).

| Claims | Boehm |
|---|---|
| 1. A computerized method of saving version and product information of at least one library in an executable program, comprising:<br>(1) creating a **version source file** for the at least one library, the at least one library having a **name** and containing one of more functions and the version source file containing **version** and product information pertaining to the at least one library; | Boehm's invention has taught us a method and apparatus for building a software system, in SUMMARY, line 43-46, "the invention lists each source file, by **name** and **version number**, (the **name** can be the **product information**) and each explicit object file (by name) that comprise the software system to be built and form that **build list**..." Line 50-52, "Source file links associate **source file names** and **versions** listed on the **build list** (here the **build list** is same as the **version source file** stated in claim 1 first item) to a copy of the corresponding source file store in the network cache memory". |

| | |
|---|---|
| (2) compiling the version source file to create a version object file;<br>rebuilding the at least one library to include the version object file; and | In Boehm's column 13, line 1-3, "the source file handler 207 generates the potentially usable object filename 'Adder.o' from the source file 'Adder.src,'" this shows the **version source file** has been compiled into the **version object file** as recited in second item of claim 1. It also mentioned, "**rebuild** object files that may be changed from the prior builds" (see Boehm's Abstract). – In order to make a new executable, the library would have to be rebuilt with the newly compiled object code. |
| (3) building the executable program to include the at least one library such that the version and product information in the version object file is combined into the executable program. | In Boehm's column 8, line 35-40, "After the object file handler processes the explicit object files listed on the build list 100 and the potentially usable object files generated by the source file handler, **the program can be built** using *make* or another software-building program (shown as step 300 in Fig. 2)." The *make* tool compiles given **source files** to **object files** and link them all together to form an **executable program** therefore all the version and **product information** (file name, date and time of the build, build name, ) is |

and time of the build, build name....) is combined into the executable program.

Boehm shows the flow of creating a build list of versioned resource files; compiling the versioned source files to create versioned object files, rebuilding a new object file list, and then building a new executable program (see Fig. 4, 5, 7A and 7B). Therefore, Boehm's inventions covers entire claim 1.

7. A method of saving as recited in claim 1, wherein the step of rebuilding the library includes:

removing any version object file from the library; and remaking the library to include the version object file.

In Boehm's invention, column 1, line 25-29, "... efficient software design, by providing a **library** or **storehouse** of previously developed software items and modules that can be re-used, modified, expanded, or reconfigured as required to support development of upgrades, new applications or new systems." Boehm teaches '**a library** or **a storehouse**', which is used to reconfigure libraries as required, here the reconfiguration can be **removing** or **remaking** a library.

***Claim Rejections - 35 USC § 103***

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6457170 by Boehm et al. as applied to claims above, further in view of U.S. Patent No. 5805899 by Evans.

The explanations for each of the rejected claims are as following:

| Claims | Boehm / Evans |
|---|---|
| 2. A method of saving as recited in claim 1, wherein the step of creating the version of the source file includes: | For the features of claim 1 see Boehm. |
| (1) constructing a version string, the version string containing version and product identification information pertaining to the library; | In Boehm's SUMMARY, line 43-46, "the invention lists each source file, by **name** and **version number**"; here the **name** can include product identification information. For example, the file name Boehm used in rejection of claim 1 (2), *adder.src* can be modified to *adder_11_14_2003_v1.ada*, which shows the function of the file (it's an adder), the date (11/14/2003), the version (1), and the file is a source file in 'Ada' programming language. The version string |

| | |
|---|---|
| | can include all the **product identification information** that the user needs. |
| (2) combining function symbols with the version string to form a version function; | Boehm teaches all aspects of the applicant's claims except it does not specifically mention to associate function symbols with version name. However, Evans teaches the feature of a method and apparatus for providing a **mapfile** specifying a function name associated with a version of the software program. Evans' invention taught us that the **global symbols** and **version names** associated with each version are defined in a "mapfile" (Evans column 2, line 26-27). Here the function symbol is the global symbol, which is a symbol of a function that is known globally in the program. Therefore, it would have been obvious to a person of the ordinary skill in the art at the time of the invention to modify Evans' system with the source file versioning feature ("mapfile") for the same reason it is taught by Boehm, to provide a build list that associates the version information with the source file and based on the version information to produce versioned object and, further to build versioned executable program. |

| | |
|---|---|
| (3) creating a version source file whose name includes a keyword and the name of the at least one library; and | See the rejection of claim 1; the **build list name** can include a keyword and the name of the library. |
| (4) storing the version function in the version source file. | See the same reason as the item (2) of claim 2; the **mapfile** can include the version function name. It has to be stored in the version source file in order to be built in the executable program. |

***Claim Rejections - 35 USC § 103***

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 9, and 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6457170 by Boehm et al. as applied to claims above, further in view of U.S. Patent No. 5805899 by Evans, and further in view of U.S. Patent No. 6499137 by Hunt.

| **Claims** | **Boehm / Evans / Hunt** |
|---|---|
| 9. A method of saving as recited in claim 1, wherein there is a plurality of libraries; and | For the features of claim 1 see Boehm. |
| wherein the steps of creating, compiling and rebuilding are performed | Boehm teaches all aspects of the applicant's claims about creating build |

for each library of the plurality of libraries.

list, compiling, and rebuilding (see rejection of claim 1) executable program, but Boehm does not specifically mentioned his invention is for 'a plurality of libraries'. However, both Evans and Hunt teach the idea of including **multiple libraries** (object files) in a new build.

In Evans' invention, Column 1, 2nd paragraph, "The software development process is especially problematic when a software application binds to **shared objects (also called "libraries**"). Evans described the 'plurality of object files' as 'plurality of libraries".
In Hunt's invention, Abstract, 3rd line, "... a user selects a library for linking to the compiled application. An association is made between the **selected library and any external libraries** referenced within the compiled application."
It would have been obvious to a person of the ordinary skill in the art at the time of the invention to modify Evans' and Hunt's including a plurality of libraries for the same reason it is taught by Boehm, to provide a build list that

| | |
|---|---|
| | associates the version information with the source file (from the plurality of libraries) and based on the version information to produce versioned object files, and further to build a versioned executable program. |
| 10. A method of saving as recited in claim 9, further comprising the step of, prior to the building step, | For the features of claim 9 see Boehm in view of Evans and further in view of Hunt. |
| (1) selecting from the plurality of libraries a group of libraries needed for the building of the executable, each library in the group having a version object file; and | See the rejection of claims 1 and 9. |
| (2) wherein building the executable includes:<br>creating a temporary storage area; | In Boehm's invention, column 1, line 25-29, "... efficient software design, by providing a **library** or **storehouse** of previously developed software items and modules that can be re-used, modified, expanded, or reconfigured as required to support development of upgrades, new applications or new systems." Here **a library** or **a storehouse** is used as a **temporary storage area** for manipulating new builds. |

| | |
|---|---|
| (3) obtaining the version object file from each of the selected libraries, each version object file having a name that includes a keyword and the name of the library in which the version object file resides; | See the rejection of claims 1, 2, and 9. |
| (4) storing each of the version object files in the temporary storage area; | See the rejection of claims 2(4) and 10 (2). |
| (5) creating a list of the names of the stored version object files; and | See the rejection of claim 1. |
| (6) compiling into the executable each of the stored version object files in the list so that the executable contains any functions needed by the executable from each library in the group and the version and product information of each library of the group. | See the rejection of claims 1 and claim 9. |

7. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. US6457170 by Boehm et al. as applied to claim1 above, further in view of U.S. Patent No. 5649200 by Leblang.

The explanations for each of the rejected claims are as following:

| Claims | Boehm / Leblang |
| --- | --- |
| 8. A method of saving as recited in claim 1, wherein building the executable includes: | For the features of claim 1 see Boehm. |
| (1) creating a temporary storage area; | In rejection of claim 10 item (2), Boehm teaches the idea of keeping a 'storehouse' as a working space, where old space may be removed, and new space can be allocated (reconfigured); even Boehm's a library or a storehouse has the same function as a temporary storage area, Leblang also shows similar idea, in column 12, lines 18 –22, "View-private storage ensures that developers have enough 'elbow room' to work effectively, without getting in each others' way." Here the "**elbow room**" is used as a **temporary storage space**. It would have been obvious to a person of the ordinary skill in the art at the time of the invention to include Leblang's idea about keeping 'elbow room' to make work effectively for the same reason it is taught by Boehm to provide enough space, to reconfigure the space as required to support development of upgrades, new application, or new systems. |
| (2) obtaining the version object file from | Same reason as rejection of claims 1 and |

| | |
|---|---|
| the library, the version object file having a name that includes a keyword and the name of the library in which the version object file resides; | 2 (3). |
| (3) storing the version object file in the temporary storage area; and | Same reason as rejection of claims 2 (4), 10(2) and 8(1). |
| (4) compiling into the executable the stored version object file so that the executable contains the version and product information pertaining to the library. | Same reason as rejection of claims 1 and 2(1). |

8. Claims 3, 4-6, 11-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6457170 by Boehm et al. as applied to claims above, further in view of the "Software Engineering", by Stephen R. Schach, 1990, and further in view of a UNIX reference "Separate Compilation and the UNIX **make** Program" by Gary Shute, 09/23/1999.

The explanations for each of the rejected claims are as following:

| **Claims** | **Boehm / Schach / Evans / Shute** |
|---|---|
| 3. A method of saving as recited in claim 1, wherein the step of creating the version source file includes: | For the feature of claim 1 see Boehm. |

| | |
|---|---|
| constructing a version string, the version string containing version and product identification information pertaining to the library; | See rejection of claims 1 and 2(1). |
| combining a build identifier and function symbols with the version string to form a name of a build-identified version function; | See rejection of claim 2(1). Boehm has mentioned using existing build tool, for example, the *make* tool (rejection of claim 1 (3)). Boehm has taught us to use the make tool, but didn't mention the detail of how to specify source file name, object file name and assign a build name. The *make* command is described in detail in the Separate Compilation and the UNIX make Program (by Gary Shute, 09/23/99), page 4, it shows the way of specifying these names. The example was given on page 4, "*table.O : table.C table.h*", where to enter source file names (*table.C* and *table.h*) and to produce an object file (*table.O*); and to create a build by "*calculator : calculator.o table .o*" (using object files *calculator.o* and *table.o* to create a build named *calculator*). Note that the any function symbols or the version string can also be used here; user can combine a build identifier and function symbols with the version string to form a **build-identified version function** that the |

| | |
|---|---|
| | user wants to be included in the new build.<br><br>It would have been obvious to a person of the ordinary skill in the art at the time of the invention to utilize the *makefile* concept for the same reason it is taught by Boehm, to combine a build identifier and function symbols with the version string to create a version source file. |
| creating a version source file whose name includes a keyword and the name of the library; and | See the rejection of claim 2 (3). |
| storing the build-identified version function in the version source file. | See the rejection of claim 2 (4). |
| 4. A method of saving as recited in claim 3, wherein the build identifier is a **date** on which the build occurs. | For the features of claim 3 see Boehm, Schach, Evans, and Shute.<br>Boehm teaches all aspects of the applicant's claims except he does not specify how to name the build identifier. However, Schach teaches the feature of what to be included in a build name so it can uniquely identify builds. In Schach's "Software Engineering", section 11.5.1, page 353, 3rd paragraph, "... a detailed |

record (or derivation) of every version of the product must be kept. This comprises the **name of each source code element**, **including the variation and revision** (as recited in claim 5**, build identifier**), the **versions** of the various compilers and linker used, the **name of the person** (as recited in claim 6**, a user that performs the build**) who constructed the product, and, of course, the **date and the time** (as recited in claim 4, **date**) at which it was constructed." This paragraph basically covers claims 4, 5, and 6.
It would have been obvious to a person of the ordinary skill in the art at the time of the invention to include Schach's idea about uniquely identify builds for the same reason it is taught by Boehm, to provide a build list that associates the version information with the source file and based on the version information to produce versioned object and further to build versioned executable program.

See rejection of claim 4.

5. A method of saving as recited in claim 3, wherein the **build identifier** is a **number** that uniquely identifies the build.

See rejection of claim 4.

| | |
|---|---|
| 6. A method of saving as recited in claim 3, wherein the build identifier is an identifier of **a user that performs the build**. | |
| 11. A method of saving as recited in claim 9, further comprising the steps of: prior to the building of the executable, | For the features of claim 9 see Boehm in view of Evans and further in view of Hunt. |
| (1) selecting a group of libraries from the plurality of libraries, each library in the group having a version object file; and | Boehm teaches the entire flow of selecting group libraries, building the object files, which includes version source files, and build executable with the entire versioned object files. But Boehm does not specifically mention building a compound library. However, in Schach's "Software Engineering", section 11.6.5, page 359, 3rd paragraph, "For each executable load image, the programmer sets up a *Makefile* specifying the hierarchy of source and object files that go into that particular configuration; such a hierarchy is shown in Figure 11.2." Figure 11.2 on page 352 is very important, it shows the way selecting different groups of libraries (object files), binding object files into a compound library (executable load image), and the new executable program is stored within the compound library. |

| | |
|---|---|
| | Schach teaches the feature of combining various libraries into a compound library. The UNIX *make* program is an existing build tool which builds new executable program from given versions of source file/object files within different libraries. It would have been obvious to a person of the ordinary skill in the art at the time of the invention to include Schach's idea about making an executable from the compound library and using the *make* command, for the same reason it is taught by Boehm, to provide a build list that associates the version information with the source file, and based on the version information to produce versioned object files, and further to build versioned executable program. |
| (2) building a compound library from the selected group of libraries, the compound library including a version object file for the compound library and the version object files of each library in the group; and | See the rejection of claim 11 (1). |
| (3) wherein the step of building the executable includes building an executable | See the rejection of claim 11 (1). |

to include the compound library, such that the version and product information of the compound library and each library in the selected group are combined into the executable program.

| | |
|---|---|
| 12. A method of saving as recited in claim 11, wherein the step of building a compound library includes: | For the features of claim 9 see Boehm in view of Evans and further in view of Hunt. |
| creating all object files, including the version object files, from each library of the selected group; | See the rejection of claims 1 and 11(1). |
| extracting all object files, including the version object files, from each library of the selected group; | See the rejection of claim 11 (1). |
| storing the extracted object files for all libraries of the selected group in the temporary storage area: | See the rejection of claims 11 (1) and claim 8 (1). |
| creating a version source file for the compound library, the version source file for the compound library containing version and product information pertaining to the compound library; | See the rejection of claims 2(3) and 11 (1). |
| compiling the version source file to create a version object file for the | See the rejection of claim 11 (1). |

compound library;

| | |
|---|---|
| storing the version object file in the temporary storage area; | See the rejection of claims 2(4), 8 (1) and 10(2). |
| building the compound library from the object files stored in the temporary storage area; | See the rejection of claims 8 (1) and 11(1). |
| saving he compound library in the library storage area; and deleting the temporary storage area. | See the rejection of claims 10 (2) and 8 (1). |
| 13. A method of saving as recited in claim 9, further comprising the steps of: prior to the building of the executable, | For the features of claim 9 see Boehm in view of Evans and further in view of Hunt. |
| selection a group of libraries from the plurality of libraries, the group including at least one compound library and each library in the group having at least one version object file; and | See the rejection of claim 11 (1). |
| building a multiple compound library from the selected group of libraries, the multiple compound library including a version object file for the multiple compound library and the version object files of each library in the group; and | See the rejection of claim 11 (1). |

| | |
|---|---|
| wherein the step of building the executable includes building an executable to include the compound library, such that the version and product information of the multiple compound library and each library in the selected group are combined into the executable program. | See the rejection of claims 1, 2, and 11. |

***Conclusion***

9. The following summarizes the status of all the claims:

*102 (e) rejections:* 1, 7

*103 rejections*: 2 – 6, 8 – 13

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 703-305-7205. The examiner can normally be reached on 6:30am to 3:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on 703-305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-308-3988.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

Chih-Ching Chow
Examiner
Art Unit 2122

CC

TUAN DAM
SUPERVISORY PATENT EXAMINER